

## AMENDMENT TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application.

### Listing of Claims:

1. (currently amended) A system for managing coherent data access through multiple nodes, comprising:

a first data processing system forming a first node, in which the first data processing system includes a first processor, a first bridge, a first interface and a first memory that is local to the first node, wherein the first node supports packet traffic for transfer of packets, coherent traffic to access ~~local and remote~~ the first memory and non-coherent traffic to communicate with input/output (I/O) circuitry, in which the first memory stores cacheable data having global coherency; and

a second data processing system forming a second node that also supports packet traffic, coherent traffic and non-coherent traffic, in which the second data processing system includes a second processor, a second bridge, ~~and~~ a second interface and a second memory that is local to the second node and also having global coherency, the first and second interfaces coupling the first node to the second node for transfer of packet traffic, coherent traffic and non-coherent traffic between the first and second nodes, wherein when the second node receives a packet from an external source that is to access a coherent fabric of the first memory, the second bridge identifies the ~~first node as~~ memory to be located in a remote node and converts the packet ~~as to~~ to an uncacheable data access request to the first node so that the ~~uncacheable data access request~~ access to the first memory does not access a coherent fabric ~~of the second memory~~ in the second node, and when the first bridge receives the uncacheable data access request, the first bridge identifies the uncacheable data access request as a local access request to the first memory in the first node and processes the uncacheable data access request from the second node as a coherent data access request in the first node to access the coherent fabric of the first memory in the first node.

2. (canceled)

3. (currently amended) The system of claim 1, wherein the uncacheable data access request associated with the packet from the external source is a store or a write request to access the first memory.

4-5. (canceled)

6. (previously presented) The system of claim 1, wherein the uncacheable data access request by the second bridge follows a producer-consumer protocol.

7. (currently amended) The system of claim 6, wherein a payload and a flag are written to a memory location in a home node when following the producer-consumer protocol.

8-9. (canceled)

10. (currently amended) A method for managing coherent data access through multiple nodes, comprising:

establishing a cacheable coherent memory space in a ~~local~~ first memory of a first data processing system that forms a first node, the first memory being local to the first node and to store cacheable data having global coherency, wherein the first node that supports packet traffic for transfer of packets, coherent traffic to access ~~local and remote~~ the first memory and non-coherent traffic to communicate with input/output (I/O) circuitry, and in which the first data processing system also includes a first processor, a first bridge and a first interface;

receiving at a second node a packet from an external source to access a coherent fabric of the first memory in the first node, wherein the second node also supports packet traffic, coherent traffic and non-coherent traffic and is formed of a second data processing system that includes a second processor, a second bridge, ~~and~~ a second interface and a second memory that is local to the second node and also having global coherency, the first and second interfaces coupling the first node to the second node for transfer of packet traffic, coherent traffic and non-coherent traffic between the first and second nodes;

identifying in the second node that the first memory that is to be accessed is located in a remote node;

converting the packet in the second bridge ~~as to~~ to an uncacheable data access request to access the first memory in the first node, so that ~~the uncacheable data access request access to the first memory~~ does not access a coherent fabric of the second memory in the second node;

transferring by the second bridge the uncacheable data access request to the first node;

receiving by the first bridge the uncacheable data access request from the second node through coupling between the first and second interfaces;

identifying the uncacheable data access request to the first memory as a local access ~~to the memory~~ in the first node;

processing the uncacheable data access request from the second node as a coherent data access to access the coherent fabric of the first memory in the first node; and

accessing the coherent fabric of the first memory in the first node in response to receiving the packet from the external source, but without accessing the coherent fabric of the second memory in the second node.

11. (canceled)

12. (currently amended) The method of claim 10, wherein the uncacheable data access request associated with the packet from the external source is a store or a write request to access the first memory.

13-14. (canceled)

15. (previously presented) The method of claim 10, wherein the uncacheable data access request by the second bridge follows a producer-consumer protocol.

16. (currently amended) The method of claim 15, wherein a payload and a flag are written to a memory location in a home node when following the producer-consumer protocol.

17-18. (canceled)